NAVAL POSTGRADUATE SCHOOL
MONTEREY, CALIFORNIA

THESIS

A STOCHASTIC ENHANCEMENT TO THE ANALYST'S WORKBENCH

By

Andrew W. Melton

September 1996

Thesis Advisor: Gary R. Porter

Approved for Public Release, Distribution is Unlimited

REPORT DOCUMENTATION PAGE			Form Approved OMB No. 0704-0188
Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instruction, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188) Washington DC 20503.			
1. AGENCY USE ONLY *(Leave blank)*	2. REPORT DATE September 1996	3. REPORT TYPE AND DATES COVERED Master's Thesis	
4. TITLE AND SUBTITLE **A STOCHASTIC ENHANCEMENT TO THE ANALYST'S WORKBENCH**			5. FUNDING NUMBERS
6. AUTHOR(S) Andrew W. Melton			
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Naval Postgraduate School Monterey CA 93943-5000			8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)			10. SPONSORING/MONITORING AGENCY REPORT NUMBER
11. SUPPLEMENTARY NOTES The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.			
12a. DISTRIBUTION/AVAILABILITY STATEMENT Approved for public release; distribution is unlimited.			12b. DISTRIBUTION CODE

13. ABSTRACT *(maximum 200 words)*

The Analyst's WorkBench is a deterministic intergrated framework developed and used by the Weapons Planning Group at NAWC China Lake. The model has no stochastic capability which requires all analysis to be conducted using parameters based on expected values of occurrence. This thesis develops a stochastic enhancement that can be incorporated in the Analyst's Workbench. Independent identically distributed (IID) events can be generated by calls to the enhancement as a parametric input. To demonstrate the application of a stochastic process within the Analyst's WorkBench, a test scenario of a ship defense model is developed. A large scale missile attack is simulated deterministically and stochastically to demonstrate the differences of a random probability of successful defense vice an expected value of success. It is shown that the stochastic results provide a more realistic simulation and that the deterministic results overstate the capability of a system subject to random events that can be described by a statistical distribution.

14. SUBJECT TERMS WorkBench			15. NUMBER OF PAGES 66
			16. PRICE CODE
17. SECURITY CLASSIFICATION OF REPORT Unclassified	18. SECURITY CLASSIFICATION OF THIS PAGE Unclassified	19. SECURITY CLASSIFICATION OF ABSTRACT Unclassified	20. LIMITATION OF ABSTRACT UL

NSN 7540-01-280-5500

Standard Form 298 (Rev. 2-89)
Prescribed by ANSI Std. 239-18 298-102

Approved for Public Release, Distribution is Unlimited

A STOCHASTIC ENHANCEMENT TO THE ANALYST'S WORKBENCH

Andrew W. Melton
Lieutenant Commander, United States Navy (Retired)
B.S. Texas A&M University, 1979

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN OPERATIONS RESEARCH

from the

NAVAL POSTGRADUATE SCHOOL
September 1996

Author: Andrew W. Melton

Andrew W. Melton

Approved by: Gary R. Porter

Gary R. Porter, Thesis Advisor

M. P. Bailey

Michael P. Bailey, Second Reader

Frank C. Petho

Capt. Frank Petho, Chair, Department of Operations Research

ABSTRACT

The Analyst's WorkBench is a deterministic integrated framework developed and used by the Weapons Planning Group at NAWC China Lake. The model has no stochastic capability which requires all analysis to be conducted using parameters based on expected values of occurrence. This thesis develops a stochastic enhancement that can be incorporated in the Analyst's WorkBench. Independent identically distributed (IID) events can be generated by calls to the enhancement as a parametric input. To demonstrate the application of a stochastic process within the Analyst's WorkBench, a test scenario of a ship defense model is developed. A large scale missile attack is simulated deterministically and stochastically to demonstrate the differences of a random probability of successful defense vice an expected value of success. It is shown that the stochastic results provide a more realistic simulation and that the deterministic results overstate the capability of a system subject to random events that can be described by a statistical distribution.

DISCLAIMER STATEMENT

The views expressed in this thesis are those of the author and do not reflect the official policy of position of the Department of Defense or the U.S. Government.

TABLE OF CONTENTS

I. INTRODUCTION

This chapter is a review of the Analyst's WorkBench and discusses the motivation for developing a stochastic enhancement that could be incorporated within the model. The organization of the remainder of the thesis is presented at the end of the chapter.

A. THE ANALYST'S WORKBENCH

The Analyst's WorkBench (AWB) is an integrated framework for the interactive application of models and tools by the warfare analyst or fleet operator that was originally developed for use by analysts in NAWCWPNS Weapons Planning Group at China Lake California for Strike, AAW and ASuW analyses. It is a highly flexible and extensible environment that can be applied across a broad set of analytical applications. It allows the analyst to step through complex scenarios, pausing at times, or events, to utilize a variety of analysis tools and models. The Analyst's WorkBench has an extensive graphical user interface which provides the user the ability to dramatically visualize the analysis as well as the capability to document analyses directly into presentations or documents.

The Analyst's WorkBench has been used primarily at China Lake in a variety of applications. Initially planned as a replay tool for Weapons and Tactics Evaluation Center (WEPTAC) games, it is used to allow analysts an easy method to interact with scenarios and play "what-if?" excursions. It has been used to model Integrated Air Defense systems (IADS) to allow detailed investigation of strike scenarios. Its most recent application has been in a force mix study which draws heavily on efforts outside of the model to provide parametric inputs. It has

been used in US Navy fleet applications for quick verification of strike planning.

The Analyst's WorkBench is highly modular in concept. The intent of the architecture is to facilitate analysts' development of highly specialized models to populate a library for future analysts to draw on. The Analyst's WorkBench was developed with a core set of models and intended to be grown with new models as the uses of the Analyst's WorkBench became more evident in the analytical community. A model can be developed for a specific or generic study. The analyst can select models from the library to support the study being conducted. The core framework of the model contains a simulation clock that can be interactively controlled, an event queue that can be modified on the fly, a graphic interface that can take DMA data to provide a realistic display as well as allow spherical or XY coordinate systems and a variety of measurement tools to give analysts immediate access to information in the midst of a simulation.[Ref. 1]

The analyst's model is "plugged into" the Analyst's WorkBench framework by way of Model Interface GuidelineS (MIGS). These MIGS allow the analyst to use only those portions of the Analyst's WorkBench which are germane to the study resulting in improved computer utilization. Parametric inputs are provided through a scripting function called AWSUM (Analyst's WorkBench Script of User-defined Movements) and models can add commands to the AWSUM script via the MIGS architecture. The advantage of the MIGs "hooks" are that as a model is developed, the overhead is significantly reduced because most of the common code can be called from the main model. Currently the model contains seven MIGs "hooks" that provide an ability for common events to be evaluated first as a computer system event, then as an Analyst's WorkBench event and finally as a model event.

Key to the extensibility of the Analyst's WorkBench is the ability to switch

modes from a scripted depiction of a scenario to an interactive "what-if" analysis. As the name implies, Analyst's WorkBench is intended to be a tool that supports the analyst in a study effort and adapt to the study needs rather than scoping a study to adapt to the features of Analyst's WorkBench.

The Analyst's WorkBench is written as an Object Oriented Program in Think™ Pascal and runs in a Macintosh environment. It can run on any Macintosh capable of 32-bit addressing although performance is obviously enhanced when run on 68040 processors or on a PowerPC CPU. The complexity of the models that are plugged into the framework affect the amount of RAM actually required, but in general the Analyst's WorkBench will run reasonably well in an 8 MB environment.

Analyst's WorkBench currently has no ability to simulate a stochastic process. In its current employment, expected values are fed into the model and single runs are utilized to test scenarios. This prevents the examination of the effects of randomness of various components in systems unless multiple runs are conducted and the input values are systematically varied. Even this will not adequately capture the effect of random events because the expected value implies that all probabilistic encounters will be treated equally. Clearly this prevents examination of queues in targeting and radar tracking applications.

The work of this thesis is to add a stochastic enhancement to the Analyst's Workbench. A tool set of common statistic distributions was written that can be called from the Analyst's WorkBench. To see the effect of stochastic modeling on a scenario similar to typical tasking of the Analyst's WorkBench, an Anti-Air Warfare / Anti-Ship Missile Defense scenario was built. It contains a large, multiple asset threat and a defense-in-depth system that is sufficiently stressed to require queuing to be examined.

A model was built that runs within the Analyst's WorkBench framework. It runs multiple runs of the same scenario with some of the input parameters randomized to allow demonstration of the stochastic enhancement that was added to the Analyst's WorkBench. Specifically, range and probability distribution functions (pdf) for various weapon events were randomized and the effect of this randomization was measured against a deterministic formulation.

The focus of this thesis is to describe the tools built to allow stochastic modeling within the Analyst's WorkBench architecture and to describe the Anti-Ship Missile Defense Model built to demonstrate a use of a stochastic method. A typical scenario will be shown from both a stochastic and deterministic standpoint and compared. From this effort conclusions will be drawn as to the value of stochastic modeling within Analyst's WorkBench.

B. THESIS ORGANIZATION

The remainder of this thesis is organized as follows:

1. Chapter II discusses the process used to develop a stochastic plug-in, the random generator utilized within the enhancement, and other statistical distributions that are supplied with the enhancement that can be used when specific distribution of random variables are desired.
2. ChapterIII discusses the scenario and the model that was developed to demonstrate the employment of a stochastic process within the simulation. A baseline scenario and excursions to the baseline are presented.
3. Chapter IV is a presentation of the results achieved from running Analyst's WorkBench using the model developed for this thesis including analysis of the ability of the stochastic simulation to produce statistically random parameters.

4. Chapter V presents the conclusions and recommendations of the thesis research.

II. STOCHASTIC PLUG IN

The most important aspect of building the stochastic enhancement was to build the statistical package from which random variables could be drawn. Law and Kelton [Ref. 2] provide a lengthy discussion of the types of statistical distributions that one might draw from in order to translate a real world model into a theoretical simulation. Selected algorithms demonstrated in the reference were developed into Pascal code and then incorporated as a unit in the Analyst's WorkBench model. This represents the majority of the coding effort that occurred in this thesis and is discussed in detail below.

A. RANDOM NUMBER GENERATOR

Modelers are well aware that random number generators are not random at all and in fact do not want them to be in order to be able to reproduce results. Reference 2 presents ample discussion of the Linear Congruential Generators (LCG) as a means to produce pseudo-random numbers for use in describing variables. Once an LCG generates an output that appears as uniformly distributed between 0 and 1, additional statistical distributions can be easily produced. Clearly, the properties of the random number generator need to be rapidly generated variables that have uniform appearance without undue correlation and be reproducible. It is important to have a long period so that the simulation can be run long enough without risk of repeating numbers. Much discussion is given to the advantages of using *prime modulus multiplicative LCGs* and in determining the type of random number generator to use, it seemed unnecessary to develop a new LCG. In short, the Marse-Roberts code utilized in Reference 2 was sufficient to

provide very adequate random number generation.

As previously stated, the Analyst's WorkBench is a Macintosh computer based model and the code is written in *Think Pascal™* [Ref. 3]. *Think Pascal™* does contain a random number generator but in order to (1) ensure portability to other platforms and (2) ensure adequate control of the random number generation, it was decided to utilize the Marse-Roberts random number generator presented in Law and Kelton. This code requires a 32-bit computer architecture to run properly and some older Macintosh platforms are only 16-bit architecture. However, the Analyst's WorkBench is also based on the 32-bit architecture and this was determined to be an acceptable solution. The LCG is described in Reference 2 and follows the formula:

$$Z_i = (a \cdot Z_{i-1} + c)(\boldsymbol{mod}\,\mathrm{m}) \qquad \mathbf{2.1}$$

where c=0;

a= 630,360,016;

and m= 2,147,483,647.

The U(0,1) distribution is produced from $U_i = Z_i/m$. The Marse-Roberts code provides the generation of the Uniform random numbers as part of the code "to avoid distortion and machine-dependent variation due to the method of computing and storing floating point numbers."[Ref. 2] This random number generator is widely used and the empirical chi-square, serial and runs tests were well presented in the reference so independent testing to verify the appearance of a Uniform random number was not necessary. Figure 2-1 shows a scatter plot of 1000 random numbers generated from the first seed value in the code. Visually, this has the appearance of randomness that would be desired in a random number

generator.

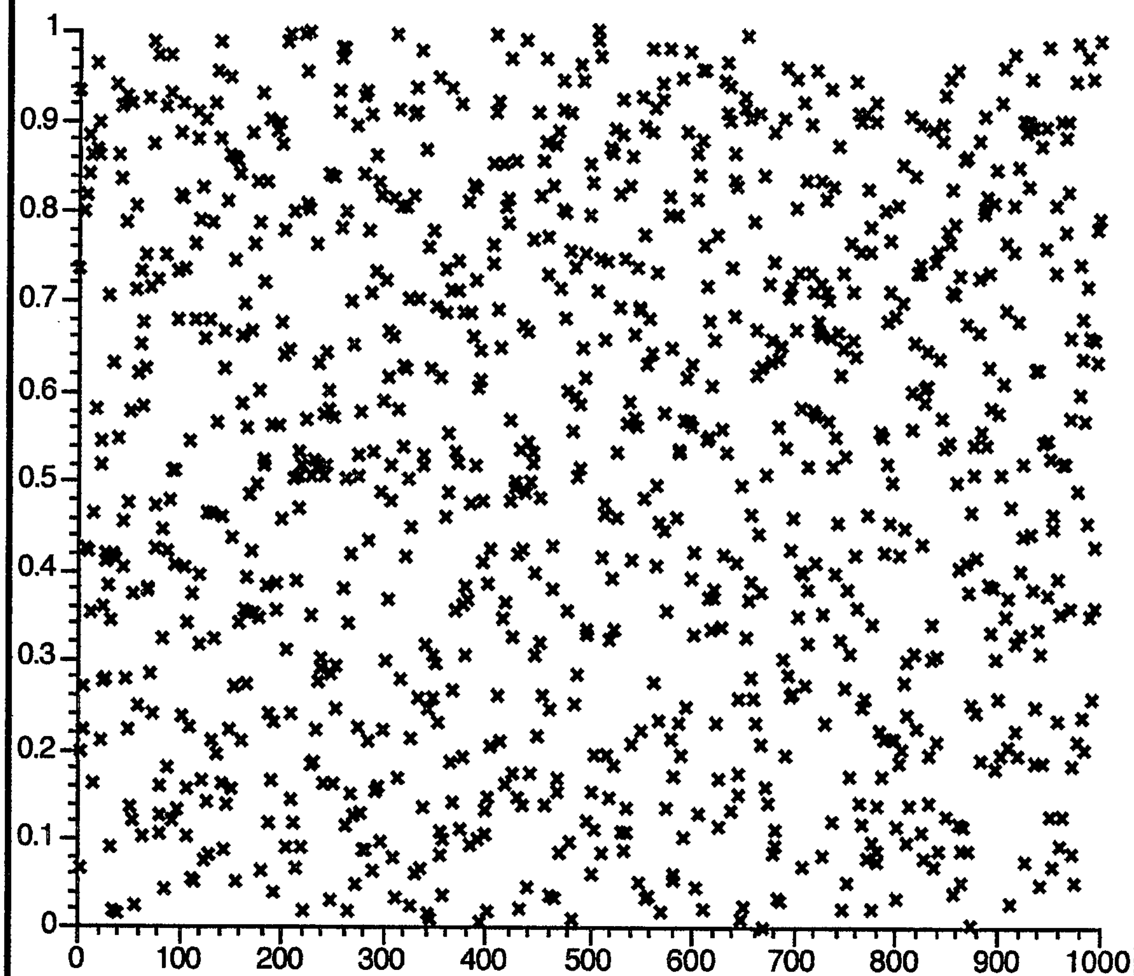

Figure 2-1: Generation of Random Numbers from Stochastic Plug-in

The feature of Marse-Roberts code that made it particularly appealing was the ability to start 100 different streams of "IID uniform random" variables by calling sequential integer seed numbers. Thus, if a variety of random variables are required as input parameters within a simulation replication, determining the seed number can be accomplished by using an integer counter rather than explicitly providing seed values.

The implementation of the generating $U(0,1)$ is accomplished either by specific calls within a specific module or via *AWSUM* scripts, when implemented.

A random number, X, distributed Uniform (0,1) is called by `rand(x)` where **x** is the integer seed value for the LCG.

B. STATISTICAL DISTRIBUTIONS

Most simulations employing the Analyst's WorkBench would probably use variables from some well known statistical distributions where only the location, scale and shape parameters need to be defined. In fact, it is envisaged that a user of the Analyst's WorkBench in a stochastic environment would either know in advance the type of statistical distribution the variables came from or want to test a hypothesis using several different distributions. Reference 2 provides a fairly comprehensive compendium of useful probability distributions that were used as the basis for importing them into the stochastic enhancement developed for the Analyst's WorkBench in this thesis. The challenge was to transform the variables generated by the LCG into random variables representing these distributions that could be called by the model. Reference 2 suggests that when generating random variables one should take into account the exactness of the generated variable, the efficiency of the generation process and the robustness of the algorithm used. It discusses different ways to generate random variables but weighs the discussion heavily in favor of using the *Inverse Transform Method* that follows the general algorithm of:

a. Generate $U \sim U(0,1)$.

b. Return $X = F^{-1}(U)$.

The algorithms presented in Chapter Eight of Reference 2 were utilized as the basis of the stochastic unit code presented in Appendix A. The algorithms as they appears in Reference 2 are reviewed here with some explanation of how an

analyst might want to employ them with in the Analyst's WorkBench. In all cases the analyst would have to provide parameters to bound the distribution; location parameters, γ, the scale parameters β, and the shaping parameters α. These parameters normally will be provided in the AWSUM script as part of the set up in lieu of a specific parameter value provided to the Analyst's WorkBench as it is presently used. An additional parameter required is a seed value. Using a variety of seed values for the generation of the initial uniform random variate will allow an additional if not arbitrary randomization facet.

1. Uniform

The Uniform Distribution is easily obtained because whenever a random number is called, a Uniform value distributed between 0 and 1 is returned. The algorithm used to generate a *U*(a,b) uniform variate is:

1. Generate U~U(0,1)
2. Return $X = a + (b - a)U$

The code for this random variable is called by `UniformN` and requires the analyst to supply a seed integer for the random number generator; and low and high real number to bracket the distribution.

2. Normal

The Normal distribution is certainly well understood and widely used in a variety of applications. One use is the simulation of munitions impact circular error probable (CEP). The algorithm to create the Normal distribution actually produces two IID Normal (0,1) random variates but only one is utilized in the call in order to simplify the scripting call in the program. The method selected to

generate Normal random variates is the polar method described by Marsaglia and Bray and is shown below:[Reference 4]

1. Generate U_1 and U_2 as IID U(0,1), let $V_i = 2U_i - 1$ for i=1,2 and let $W = V_1^2 + V_2^2$.
2. If $W > 1$, go back to step 1. Otherwise, let $Y = \sqrt{(-2\ln W)/W}$, $X_1 = V_1 Y$, and $X_2 = V_2 Y$. Then X_1 and X_2 are IID N(0,1) random variates.

It is noted [Ref. 2] this method is slower than some newer methods, but the simplicity of the method overrides the speed gains. Of note with the Normal distribution is that it can be used to generate negative values. This would not be useful in event queuing calls but can be very useful in placement decisions. If the analyst chose to use a Normal distribution for event queuing (such as determining actual launch time around a scheduled Time Of Launch) these events would have to be scheduled prior to running the event queue or inserting some mechanism to reject negative numbers. Also, the IID normal variates that are produced do not generate a true normal distribution since neither the Normal distribution or its inverse have a simple closed form. However, this should not be a problem in the application of the Analyst's WorkBench because extreme normal variables will tend to be rejected or numerically resolved in practice.

The analyst must specify the seed number for the LCG and the location and the shape of the distribution by supplying a mean (μ) and a sigma value (σ). The random variable is then calculated by $X' = \mu + \sigma X$. The code is called by `Normalx`.

3. Exponential

The exponential distribution is very useful for modeling interarrival times between events or service times that occur at a regular rate. Initial radar detections

could be modeled by an exponential distribution with the mean describing the average time between detections.

The algorithm for generating an exponential random variable is shown below. Reference 2 makes the point that the U used in step 2 is actually $1-U$ to make the correlation between the X's and U's positive. The code does this without action of the analyst.

1. Generate $U \sim U(0,1)$

2. Return $X = (-\beta)\ln U$

The call to generate an exponential random variable is `ExponentialX`, specifying the seed value and a real ß value.

4. Gamma

The shape of the Gamma distribution curves suggests a possible use to describe time to complete a task. The scale and shape parameters $\alpha\,\beta$ describe the mean time to complete the task. There can never be negative or zero values.

Reference 2 notes that $X' \sim$ gamma(α,β) is merely $\beta(X \sim$ gamma $(\alpha,1)$ provided $\beta > 0$ and that the gamma (1,1) is the exponential distribution with mean 1. This lead to the algorithm that has three different cases dependent on the value of α; $0 < \alpha < 1$, $\alpha = 1$ and $\alpha > 1$. In the case where $0 < \alpha < 1$, the algorithm specified in reference 2 as follows:

1. Generate $U_1 \sim U(0,1)$ and let $P = bU_1$. If $P > 1$, got to step 3. Otherwise proceed to step 2.

2. Let $Y = P^{1/\alpha}$, and generate $U_2 \sim U(0,1)$. If $U_2 \leq e^{-Y}$, return X= Y. Otherwise, go back to step 1.

3. Let $Y = -\ln[(b - P)/\alpha]$ and generate $U_2 \sim U(0,1)$.If $U_2 \leq Y^{\alpha 1}$, return X= Y. Otherwise, go back to step 1.

In the case where $\alpha > 1$, Reference 2 describes an algorithm developed by Cheng which use constants that were incorporated in this code as follows:

1. Generate U_1 and U_2 IID$U(0,1)$.
2. Let V= $a \ln\left[\frac{U_1}{(1-U_1)}\right]$, $Y = \alpha e^V$, $Z = U^2{}_1 U_2$, and $W = b + qV - Y$
3. If $W + d - \theta Z \geq 0$, return $X = Y$. Otherwise, proceed to step 4.
4. If $W \geq \ln Z$, return $X = Y$. Otherwise go back to step 1.

The call to generate a Gamma random variable is `GammaN`, specifying the seed value and a real α and ß value.

5. Poisson

The Poisson process is well understood as a description of interarrival times for most event queues where the probability of the next event occurring follows an exponential distribution. This process is particularly useful in modeling the rate that events are placed on a queue when the process leading to the appearance on the queue is not as important to the analysis. The call to generate a Poisson random variable is `PoissionP`, specifying the seed value and a real λ value.

6. Weibull

The Weibull distribution can be used in reliability studies to specify the task completion times or equipment failure times. A Weibull distribution is useful in those instances where negative values about the mean are not desired. The Weibull distribution inverse-transform algorithm:

1. Generate U IID$U(0,1)$.
2. Return X= $\beta(-\ln U)^{1/\alpha}$

The call to generate a Weibull random variable is `Weibull`, specifying the seed value and a real α and β value.

7. *m*-Erlang

Many stochastic processes are described as following an Erlang distribution. The m-Erlang distribution is achieved by using a convolution algorithm to sum a series of IID exponential random variables with the same mean. The call to generate a m-Erlang random variable is `mErlang`, specifying the integer seed value, the integer m, and a real β value.

8. Cauchy

The Cauchy distribution is symmetric around zero, bell shape, but has considerably heavier tails than the normal distribution. The Cauchy distribution may be used similarly to the Normal but because of the heavy tails, no moments exist in the Cauchy. The mean of a collection of IID Cauchy variables is no better an estimator of the central tendency of the distribution than any of the individual observations. The call to generate a Cauchy random variable is `CauchyX`, specifying the seed value and a real median and β value.

C. ADDITIONAL STOCHASTIC PROCESSES BUILT FROM COMBINATION OF DISTRIBUTIONS

Using the technique of convolution, additional variables can be generated. The easiest method is to sum the random variables from different distributions. Reference 2 provides methodologies for generating the desired variables from combinations of distributions. The Analyst's WorkBench will generate these combinations of random numbers through custom AWSUM script calls.

III. SCENARIO FOR TESTING THE STOCHASTIC ENHANCEMENT

Once the stochastic enhancement was added to the Analyst's WorkBench, an example scenario was created to illustrate how this enhancement might be employed. As a result of this thesis research, a stochastic module can be implemented in future revisions of the Analyst's WorkBench but immediate implementation of this capability is not possible until supporting changes to the stochastic enhancements are provided in a future revision to the Analyst's WorkBench. Therefore a separate testing MIGS was implemented to run the model in a stochastic mode and to demonstrate that the stochastic mode would provide a different, more illustrative answer than would be obtained by running the Analyst's WorkBench in a deterministic mode.

A. SHIP DEFENSE MODEL

In an article that appeared in *US Naval Institute Proceedings* [Ref. 5], a scenario was proposed to demonstrate a comparison of force structure capability in the Littoral warfare environment. A flotilla of small USN fast attack craft was compared with a task force of two USN DDG-51 class destroyers to demonstrate the cost savings and force conservation that could possibly be obtained with a fast attack flotilla. In this scenario, the chief measure of effectiveness was how the force structure fared in a concentrated missile attack aimed at the on station forces. The hypothesis was that a third world nation could conceivably launch a large scale attack of 20 surface to surface anti-ship missiles against either a single DDG-51 or a forward deployed fast attack force. The intent of this scenario was to show that either force could be overwhelmed and destroyed. This scenario was the genesis

for the model developed to test the stochastic enhancement although the model was reduce in scope to represent a single ship's defense in depth against a fixed number of missiles.

The module that was developed modeled a theoretical warship with a tier defense–in–depth system that would defend against a coordinated, large scale missile attack of 20 identical surface to surface missiles. The missiles would be launched nearly simultaneously from nearly identical distances so that the missiles would arrive at the ship's defensive perimeter at near simultaneous time. Each missile would have some probability of acquiring the target ship and some probability of detonating on the ship. It was decided that if three impacts occurred on the ship, the ship would be destroyed.

The ship was designed to have an outer perimeter search radar that would alert the ship's defensive systems when an incoming missile was detected. The tracking capability of this radar was decided to be sufficient to prevent it from being overwhelmed in this scenario. An overall battle director would track and prioritize the incoming missiles and task the tiers to attack the incoming missiles as the missiles came into the tier's engagability envelope. The tier's range of defense did not overlap its adjacent tiers. Each tier had a finite number of assets to allocate against the incoming missiles and finite number of launchers that could be used to deploy these assets at a specific rate. As missiles were detected, the tiers would be tasked to target and attack the incoming missile with an interceptor missile. As a launcher fired an interceptor, it would reload and fire at the next assigned incoming missile. Figure 3-1 is a visualization of this scenario.

This scenario became an analysis of what could be characterized as an *M/G/n* [Ref. 6], *Angry Customer Queue*. The servers in each tier are the launchers that would "service" the incoming "customers" (the missiles) which could be assumed

to arrive in an IID manner. A manager (OTCManager) prioritizes each missile based on its distance to the ship and creates a queue of incoming missiles.

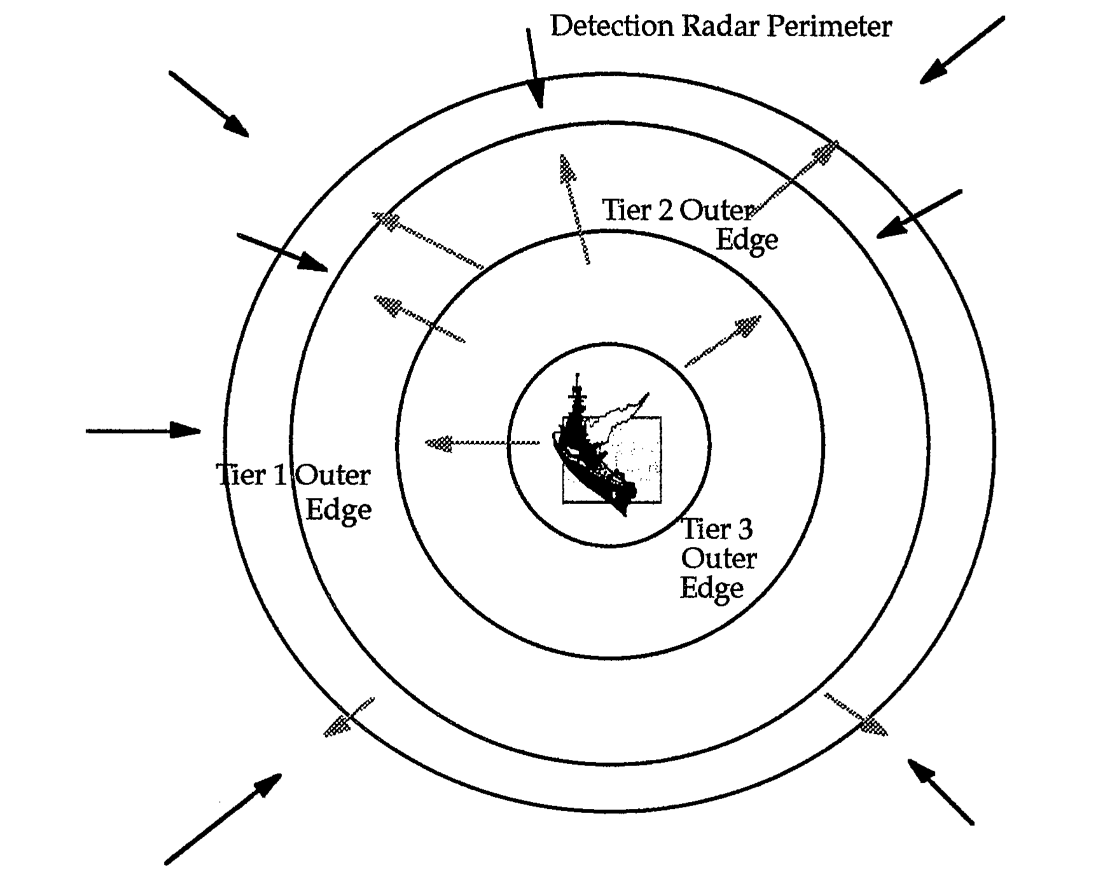

Figure 3-1: Three Tier Defense-In-Depth ASMD Model

If a tier could handle a missile but could not assign a launcher to the missile at that moment, the missile would remain in the queue.

If the missile could not be serviced by the tier before the missile traversed the tier's range of responsibility, the missile would no longer be serviced by that tier and would be assigned to the next tier. If the final tier could not service the missile within a specific time, the missile would leave the queue and be scheduled to impact the ship (hence the angry customer). An impact counter kept track of the

number of impacts and the model ran until all missiles were serviced or impacted the ship. If three missile impacts occurred prior to all missiles completing their flight, the defense tiers were assumed to be inactive and all additional service events were canceled. Servicing a missile modeled reducing the available assets to provide service with each service event and the possibility that the servicing asset would be unsuccessful in servicing (i.e., miss) the missile.

There was also the possibility the missile would miss the ship or fail to detonate which was modeled as a separate probability and had no effect on the customer queue. This is important to the model because the measure of effectiveness for this model was the ability to prevent more than three missiles from detonating on the ship. A failure to detonate would not prevent the servers from allocating an asset against the missile but would allow the model to continue to run its servers.

While it is recognized that any impact against the ship would have a potential impact on the performance of the defenses, this model did not attempt to degrade performance by impact.

This model is one dimensional; translating the missile's positional information into a distance value so that the graphical interface became unnecessary. In fact, the Analyst's WorkBench is not needed to demonstrate this model when run deterministically as the analysis can be done using rudimentary probabilities of the missile and ship characteristics. If this model were used in the Analyst's WorkBench environment, positional information and missile trajectory would be more important to demonstrate the attack more graphically. The conceptual nature of this model allows a great deal of latitude in the specification of parameters. This model could be more carefully refined to reflect more realistic behavior as desired. None of the results in this model should be construed to

provide definitive answers without further refinement. However, the model could be an excellent starting point for a more precise study of a ship's defensive capabilities using actual parameter values. In constructing this model, many of the real world ship defense characteristics where either ignored or folded into the analysis with some other parameters in order to keep the scenario simplistic enough to ease the validation of the stochastic module. For example, electronic countermeasures are an integral and significant portion of a modern warship's defense capability but rather than model that capability specifically in this module, it was decided to incorporate that into the inbound missile's probability of acquisition.

B. PARAMETERS OF THE MODEL

The parameters that effect the randomness in an object's performance are numerous and the level of model realism (and subsequent model complexity) is improved by utilizing as many of these parameters within the model as possible. Sensitivity analysis of those parameters would provide an indication of what should be actively modeled and what should be incorporated into assumptions. The intention of this thesis is to demonstrate how the model could be run stochastically so rather than establish a high fidelity model, a few choice parameters were selected to establish that the Analyst's WorkBench could work with the stochastic enhancement.

1. Missile Parameters

The attacking missiles were the objects in this model that had the most stochastic parameters. Each missile was determined to be an IID event that would have some randomness associated with it. In the case of the missile objects, the distance from the

target and the time of the missile launch were determined to be two parameters that would impact the inter-arrival times of the missiles in the attack most significantly. Speed is also something that can vary significantly but as the missiles were assumed to be identical, it was decided to hold speed constant. Table 3-1 lists the parameters that were assigned to each missile in the model. The model was constructed to allow changing these

Table 3-1 Missile Parameters

Number of Missiles	Speed	Distance from target at launch	Time of launch	P_{acq}	P_{det}
20	420 knots	~N(80nm,1nm)	~N(30secs,3secs)	0.97	0.83

parameters prior to run time but this feature was not utilized. Missile parameters remained constant between base case and excursion runs that will be discussed later in this chapter.

The distance from a target can be a function of the navigational accuracy of the launching platform, geography of the region and man made impediments such as shipping traffic. It is reasonable to assume that if a simultaneous attack was desired, the position of each missile at the time of the launch order would be equi-distant from the target. Further, it is reasonable to assume that the impact of navigational accuracy, geography and shipping would cause the position of launch of all missiles to be distributed normally around a mean value selected on the basis of the range of the missile. The thesis module uses a normal distribution with mean of 80nm and sigma of 1 nm for this scenario.

Ideally, a simultaneous launch means that all missiles leave their launchers at the same time. In practice, there is some randomness associated with the time of launch that can be attributed to factors such as electrical circuitry, booster fusing and human error. The time of launch in this study was assumed to be normally distributed with a mean *h–hour* time from the start of the model and a 3 second standard deviation. Because a negative time number is possible using this distribution, the code was written to force all times to $t \geq 0$.

The probability a missile will acquire the target is certainly the function of myriad

parameters including missile guidance characteristics, target position uncertainty, flight profile and some ship defense aspects like chaff deployment and electronic countermeasures. Playing the stochastic aspect of each of these factors would distinctly add to the realism but also adds to the complexity of the model. Thus it was decided that a single Probability of Acquisition value would be applied to this model and that it would considered a binomial distribution with each acquisition event run as a Bernoulli trial. Probability of Detonation is also considered as a Bernoulli trial. In most cases the probability of detonation is based on the reliability of the missile. In the model, random uniform numbers are generated from a uniform distribution and if the number is less than the specified number, the event is considered a success.

2. Target Parameters

The tiers were given a unique range of responsibility to prevent overlap problems. Each tier had a finite number of launchers that could launch an interceptor and recycle to launch another interceptor provided there were assets available in the tier. Each interceptor was given a Probability of Single Shot Kill (P_{SSK}) that was played as a Bernoulli trial. The interceptor's speed was held constant within each tier. Table 3-2 summarizes the characteristics of each tier. The P_{SSK} used represents an alerted and defended target.

Table 3-2: Parameters of Target Tiers

Tier	Number of Launchers	Number of Interceptors	Interceptor Speed (knots)	Max Range (nm)	Min Range (nm	P_{SSK}
1	4	20	420	40	10	0.75
2	2	10	300	10	3	0.63
3	2	50	630	3	0.25	0.45

The defense system utilized a "shoot – look – shoot" tactic. One interceptor would be allocated against the incoming missile and no other action would be taken until the results of the Intercept event were known. It allows the same tier to allocate an additional asset against an incoming missile if an intercept event can be

generated prior to the attacking missile leaving the tier's range of responsibility. This model assumes instantaneous battle damage assessment (BDA) but could be easily modified to delay or degrade the knowledge that an interceptor had missed.

C. MODEL OPERATION

This model was written to evaluate the effect of random variables on the proposed scenario. This required multiple runs of the scenario and a statistical evaluation of the results of the data collected. Correlation was determined to not be a problem in this particular model but enough data points were desired to ensure adequate population of a histogram for analysis purposes. The model was constructed so that the analyst could specify the number of runs desired and change some of the scenario parameters to allow excursions from the baseline run. The scenario parameters that could be changed for the attacking missiles (bogeys) were the position of launch (mu and sigma), the time of launch (mu and sigma), the number of attacking missiles and the missile speed. The scenario parameters that could be changed for the target tiers were the acquisition range for the ship, the maximum and minimum ranges for each tier, the number of interceptors for each tier and the P_{SSK} for each tier interceptor.

Data from each run was collected and written to output files that were then evaluated using data analysis software. The specific data that was of interest was the generation time and position of each attacking missile, and the time of arrival at the ship radar acquisition perimeter. The model collected data on the intercept events including the time and distance of the events and the success or failure of those events. Each run reported the number of impacts on the target and the number of attacking missiles killed. The final report of a scenario run was the

average number of impacts across all the runs that could then be compared to a deterministic calculation.

D. EXCURSIONS

The baseline scenario as described above was selected based on a desire to show defense in depth and the parameters selected tended to allow success of the model. It was desired to also demonstrate the effect of modification of parameters on the outcome of the scenario. Seven excursion runs were conducted to show degradation of the ships defensive capability. These excursions were to

1. Reduce the P_{SSK} of each tier's interceptors by 25 percent.
2. Reduce the P_{SSK} of each tier's interceptors by 50 percent.
3. Reduce the engagability range of each tier by 50 percent.
4. Reduce the engagability range of each tier by 75 percent.
5. Reduce the engagability range of each tier by 75 percent and anchor the range to the outer perimeter.
6. Reduce the number of interceptors in each tier by 50 percent.
7. Reduce the number of interceptors in each tier by 30 percent.

Of this list, only the engagability ranges merit explanation. In reducing the engagability range, the first two excursions were anchored to the ship; i.e., the third tier range was reduced by $x\%$ and then starting from the outer edge of tier 3's perimeter, the second tier's engagability range was reduced and so on. However, these excursions produced unexpected results so an additional engagability range excursion was run to start at the outer range, reduce its engagability range by 75 percent and then allow a gap in coverage to exist between the inner perimeter of one tier and the outer perimeter of the next tier. In this excusrion, each tier's outer

perimeter range matched the baseline case but its coverage range was reduced by 75 percent.

The baseline case and the excursions were each run 100 times. The data were collected and evaluated using a statistical software package for the Macintosh. The average runtime for each excursion was approximately 30 minutes running on a 68030 processor in 6 MBs of RAM. The data results will be presented in the following chapter.

IV. SIMULATION RESULTS

The scenario was designed to demonstrate the difference between a deterministic approach and a stochastic simulation approach to answer an analytical question. The principal measure of effectiveness in the scenario is the number of attacking missile impacts against the ship. Three impacts were deemed to constitute complete target destruction. The deterministic results can be calculated analytically without simulation as can the stochastic results but with much higher calculation complexity and chance for computational error. The stochastic results must be evaluated to demonstrate the values generated by the model follow the distribution specified and to compare the combined run results with the expected values. The excursions can be used to demonstrate that the model is sensitive to parameter variations.

A. DETERMINISTIC RESULTS

Deterministically, each of the attacking missiles was initialized at 80NM, launched at time 0 and allowed to generate events based on the set parameters specified in the previous chapter. Thus, applying the probabilities assigned to the missiles and the interceptors, a single computation of the expected probability of single shot kill (P_{SSK}) is required. The number of impacts on the ship is the sum of

$$\mathbf{P_{SSK_i} = (P_{acq}P_{det})(1-P_{Tier1})(1-P_{Tier2})(1-P_{Tier3})} \quad \mathbf{,4.1}$$

P_{SSK} for all missiles n sent against the ship thus following a binomial distribution:

$$\mathbf{Impacts = \sum_{n=1}^{20} P_{SSK_i}} \quad \mathbf{4.2}$$

While these parameters could easily have been fed into the module and run once, the simplistic calculation was accomplished on a spreadsheet.

The attacking missile P_{SSK} for the baseline case is 0.041 and the expected number of impacts is 0.82. The variance is 0.786. These are also the results for those excursions in which the only parameter that changes is the engagement range for each tier. In the case in which interceptor inventory is reduced, the expectation must be achieved by stepwise evaluation.Table 4-1 is the tabulation of results for a deterministic evaluation of the scenario.

Table 4-1: Deterministic Scenario Results

Excursion:	**Baseline**	**30% Interceptor Inventory Reduction**	**50% Interceptor Inventory Reduction**	**25% Interceptor P_{SSK} Reduction**	**50% Interceptor P_{SSK} Reduction**
P_{SSK}	0.041	0.123	0.160	0.123	0.274
Impacts	0.82	2.46	3.20	2.46	5.48

These results indicate that a 20–missile–attack as outlined above would have an expected success as shown and that additional missiles added to the attack would have additional success in a linear manner. These numbers can also be used to determine the probability that three or more missiles would impact the ship by computing the binomial distribution using the expected $\mathbf{P_{SSK}}$ and $n = 20$ as shown in Figure 4-1 below. This indicates that, using the stated parameters, the probability P(Number of Impacts$\geq$ 3) = 0.047. This probability could be used as the determination of the effectiveness of the ship's defensive system.

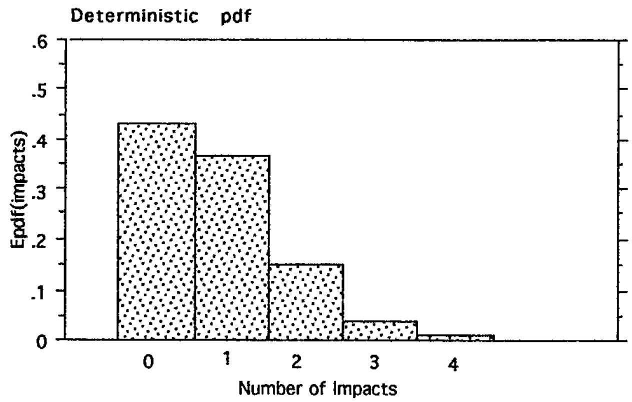

Figure 4-1: Expected pdf of Baseline case

B. STOCHASTIC SIMULATION RESULTS

This section discusses the verification used to show that the model behaved stochastically. It was desired that the missile generation time and distance should appear to be randomly distributed about a mean generation time and distance. Data was also collected to determine the distribution of the interarrival times of the missiles at the outer edge of the defense perimeter.

Running one hundred replications of the model produced 2000 missile events, which provided sufficient data to ensure statistically measurable results. The simulation is a terminating simulation in that the there were a finite number of missile events and a finite number of interceptors in each tier. The number of replications were selected to give sufficient random events to give each attacking missile IID characteristics. Each excursion was run with the same attacking missile parameters so validation of the performance of the baseline process was sufficient for the entire study. Statistical validation included standard statistical evaluation of the computed parameters. The data was collected as part of the model output

and then evaluated using StatView 4.5™ [Ref. 7], a Macintosh based statistical analysis package.

The simulation was designed to instantiate the attacking missile with a random launch time generated from a normal distribution N(30, 3). The histogram in Figure 4-2 demonstrates the launch time for the baseline model.

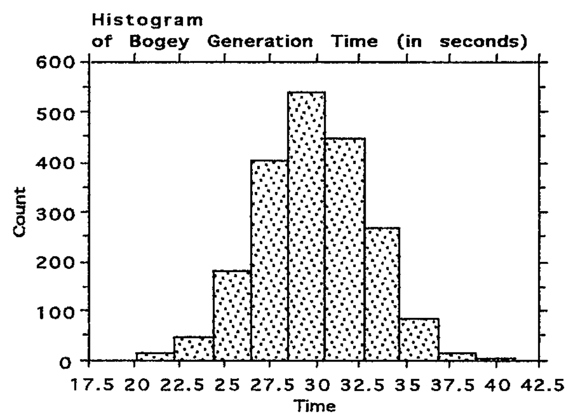

Figure 4-2: Bogey Generation Time Histogram (2000 Replications)

The attacking missile's distance from the target ship was distributed ~N(80nm, 1nm). The histogram in Figure 4-3 demonstrates the results of the replications and shows that the parameters to be distributed as desired.

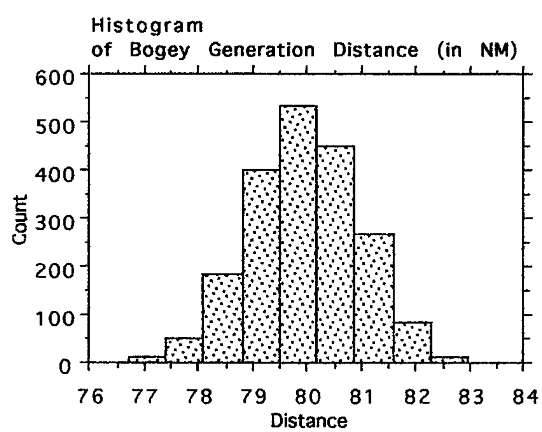

Figure 4-3: Bogey Generation Distance Histogram (2000 Replications)

The interarrival times of the attacking missiles where not modeled explicitly but the data was collected and is shown in Figure 4-4 below.

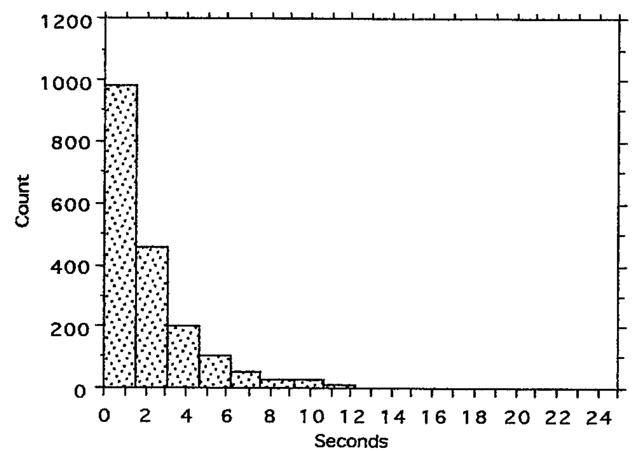

Figure 4-4: Outer Tier Bogey Interarrival Times (2000 Replications)

The missile interarrival times appear to follow an exponential distribution. This was verified by using a Q-Q plot to compare against a random series generated from an exponential distribution with the same mean as the calculated mean of the interarrival times. As shown in Figure 4-5, the Q-Q plot is fairly linear at the lower ends indicating that the interarrival time values roughly follow the exponential distribution with a mean of 2.276. This distribution could be used as the parameter for determining when missiles arrived at the outer tier of the ship's defensive system. Using the interarrival time as the stochastic parameter would eliminate the need to explicatly model the starting point of the missile. However, most current analysts are interested in the complete flight path of incoming missiles so that the actual time of detection can be measured. This necessitates the modeling of the point of origin and time of launch for an attacking missile event. The interarrival times determine queue loading for the defensive system and as such are a useful measure for analysis.

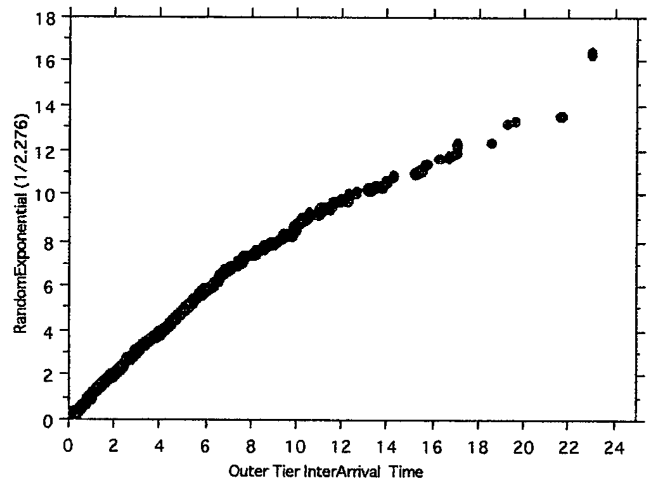

Figure 4-5 Q-Q Plot Comparison of Interarrival Time (2000 Replications) to Exponential Distribution ($\mu = 1/2.276$)

In general, the model behaved as expected in generating random variables from the appropriate distributions.

C. MODEL RESULTS

This section compares deterministic and stochastic results. First a comparison of the base cases is presented followed by an analysis of excursion cases.

1. The Base Case

Results were from the deterministic case as shown below. The results are

Table 4-2: Comparison of Deterministic Results to Stochastic Simulation

	Mean Number of Impacts	Variance of Impacts
Deterministic Results	0.82	0.786
Simulation Results	0.51	0.697

similar but the mean number of impacts is slightly different and the variance of the simulation appears less than in the deterministic case. The impact histogram for the base case (Figure 4-6) shows the relative frequency of impacts which can be interpreted as the probability distribution for the base case under the stochastic conditions at initialization.These results indicate that there is even less probability that any given 20–missile–attack will produce 3 or more impacts. In most cases, the defensive system will be sufficient to prevent any attacks from hitting the ship (i.e., P(number of impacts < 1) = 0.65) and most impacts will be less than that required to kill the ship.

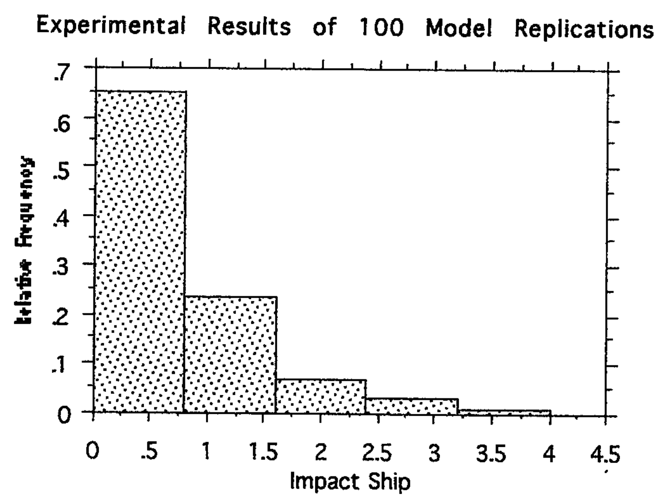

Figure 4-6 pdf Histogram of Stochastic Simulation Run (Base Case)

The simulation also provides a better understanding of the behavior of each tier's interceptors that can be readily compared to excursion cases. The base case interceptor success rate is shown below and demonstrates that the results are very close to the parameters specified at initialization. The overall success rate for the interceptors was 70.6 percent. This can be interpreted as the ship's probability of shooting down an attacking missile.

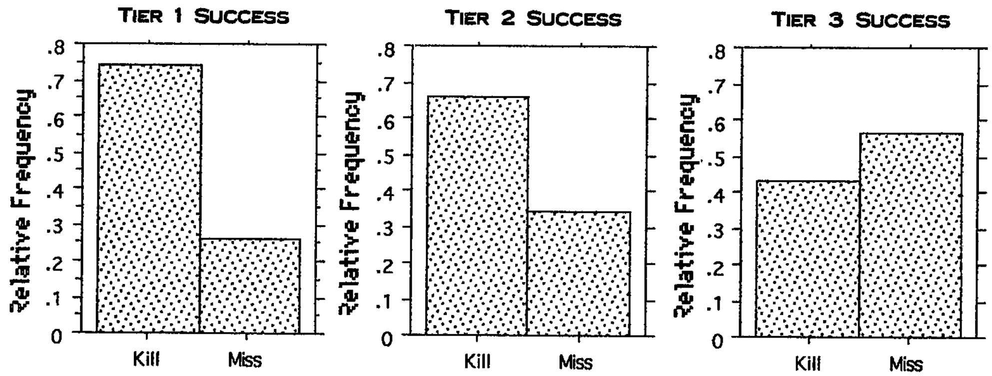

Figure 4-7 Interceptor Success by Tier

In the base case the mean number of Interceptors used was 27.46 with a variance of 7.564. Figure 4-8 shows the distribution of interceptor usage by replication. This Figure demonstrates a fairly uniform band between 22 and 34 missiles.

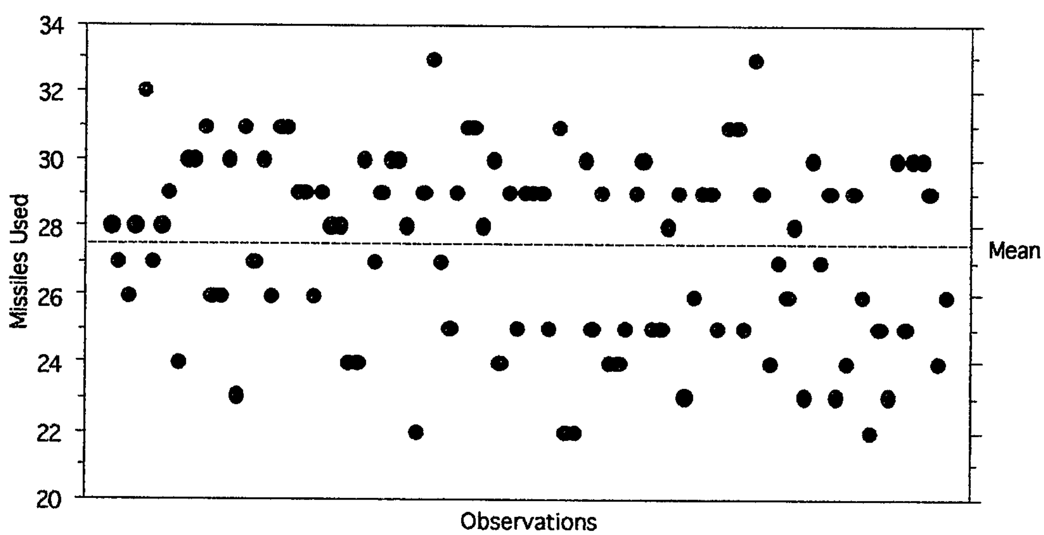

Figure 4-8: Interceptor Utilization - Baseline Simulation Analysis

2. Excursion Analysis

The excursions can be grouped into three main categories: reduction of interceptor probability of single shot kill, reduction of interceptor engagement range and reduction of interceptor inventory. The excursions in which the $\mathbf{P_{SSK}}$ of the interceptors was reduced showed the most significant increase in ship impacts. Interceptor inventory reduction also results in significant changes in ship impacts. Only the case in which engagement range was modified remained insensitive to changes. This is most likely an artifact of modeling the recycle rate of the launchers too rapidly. All attacking missiles had an opportunity to be engaged by at least one interceptor in each tier before the missile had traversed very far through the tier. Missile engagement was modeled as a uniform probability across the tier engagement range. Had the detection probability been modeled as a function of decreasing range, the model may have demonstrated more sensitivity in the reduced range excursions.

a. Reduction of Interceptor Probability of Single Shot Kill

Reducing $\mathbf{P_{SSK}}$ of each interceptor results an increased number of impacts in both the deterministic and the stochastic excursions. Table 4-3

Table 4-3: Reduction of Interceptor P_{SSK} Excursion

Excursion	Deterministic Expected Number of Impacts	Stochastic Simulation Mean Number of Impacts
Baseline Case	0.81	0.51
1/4 Reduction of Interceptor P_{SSK}	2.46	3.25
1/2 Reduction of Interceptor P_{SSK}	5.48	7.43

demonstrates that while the Baseline case has lower number of impacts in the stochastic case versus the deterministic case, the deterministic model is less sensitive to variance of the interceptor P_{SSK} parameter than the stochastic model. More P_{SSK} reduction excursions would probably demonstrate that the number of impacts on the ship would increase more rapidly in the stochastic simulation.

b. Reduction of Interceptor Engagement Range

In the deterministic case, altering the distance over which an interceptor is able to engage an incoming missile should have no effect in the expected number of impacts on the ship. No parameters that effect the probability of a missile impact are changed in this excursion. Thus in each case, one would expect the results to be the same. The mean and variance of each excursion are shown in Table 4-4. Note that while all the stochastic simulation values are

Table 4-4: Range Excursion Statistics

Excursion	Mean	Variance
Deterministic Case	0.82	0.786
Baseline Case	0.51	0.697
Half Engagement Range	0.52	0.676
One Quarter Engagement Range	0.60	0.788
One Quarter Engagement Range	0.53	0.898

generally the same, they are all less than the deterministic calculation. This indicates that the ship's defensive systems are generally more successful in defeating incoming missiles than would be indicated by a deterministic calculation. The stochastic evaluation of this excursion is shown below in Figure 4-9.

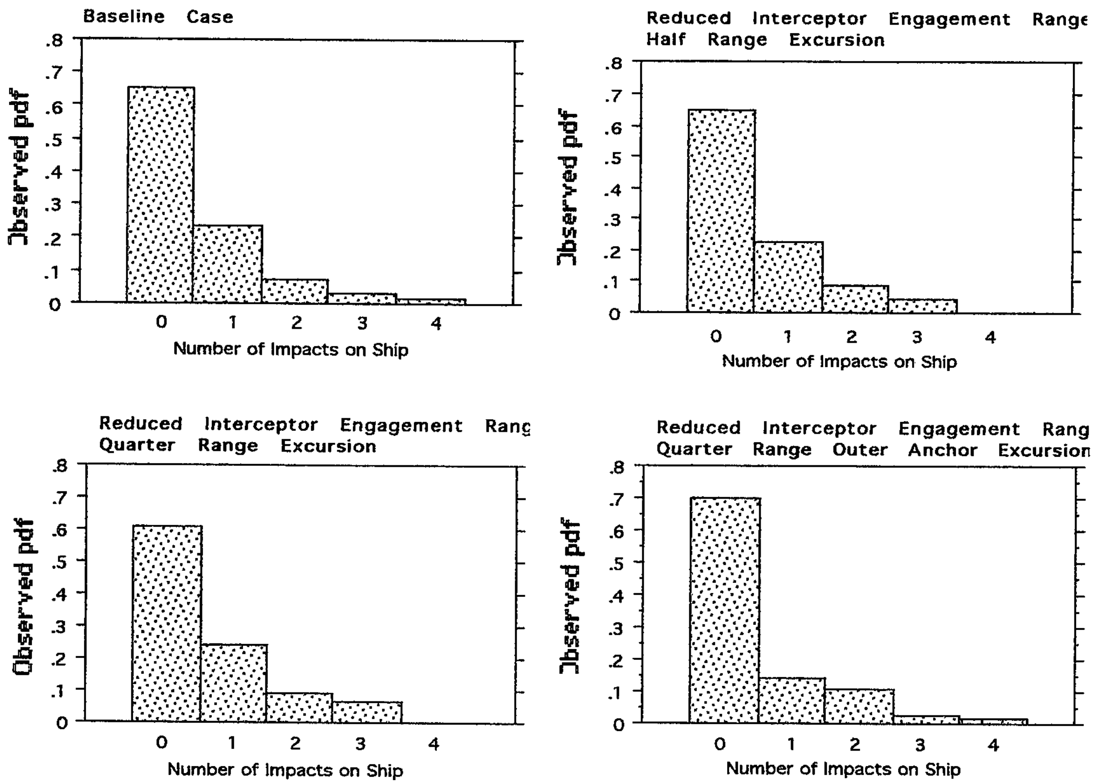

Figure 4-9 Comparison of Number of Impacts on Ship by Range Reduction Excursion

c. Reduction of Interceptor Inventory

Intuitively, interceptor inventory is significant in the overall success of the ship defense system. If the ship can not sustain a counter attack against the incoming missiles, more impacts will occur. Table 4-5 shows the comparison of the deterministic calculations and the stochastic simulation. Again the rate of change in the stochastic simulation is more rapid than in the deterministic simulation, indicating a higher sensitivity to changes in the interceptor inventory. Note that the expected number of impacts in the 50 percent inventory reduction case is 3.2. In the simulation results, the minimum number of impacts were three, which would imply under the specified scenario that there was a very low probability the

ship would escape destruction. Figure 4-10 shows the range of numbers of impacts in the excursion simulations.

Table 4-5: Reduction of Interceptor Inventory Excursion

Excursion	Deterministic Expected Number of Impacts	Stochastic Simulation Mean Number of Impacts
Baseline Case	0.81	0.51
30% Reduction of Interceptor Inventory	2.46	2.18
50% Reduction of Interceptor Inventory	3.20	5.70

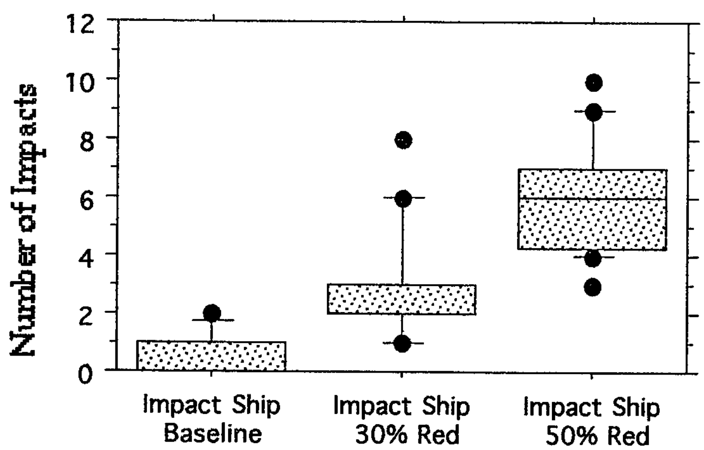

Figure 4-10: Box Plot Comparison of Simulation Results for Inventory Reduction Excursion

V. CONCLUSIONS

The history of the Analyst's WorkBench and its limitations as a deterministic model have been presented in previous chapters. The development of a plug–in that would allow the Analyst's WorkBench to produce stochastic results and a test scenario has been shown. The analysis of the results has shown that different results occur within the scenario depending on whether the model is run stochastically or deterministically. This chapter provides observations and recomendations for implementation and future research efforts.

A. OBSERVATIONS

The Analyst's WorkBench currently provides a good framework for conducting analytical studies in which the output values are deterministic. Explicit modeling of a handful of entities can be accomplished easily and their input parameters statistically varied in order to perform sensitivity analysis. Large scale modeling using changes to input parameters quickly becomes unwieldy as the number of parameters increase. Modeling this way is useful for visualization of a few parameters but is not effective for evaluating a large scale operation. The addition of the stochastic enhancement allows the Analyst's WorkBench to be modified so that a large number of random processes can be evaluated easily and visualized effectively.

Using the ship defense scenario and its excursions to generate output using the Analyst's WorkBench in a deterministic mode can be compared to its performance in the stochastic mode. In the deterministic case, for example, the only way to examine engagement activity would be to explicitly script the

engagements. Figure 5-1 shows the collection of all the engagement distances in

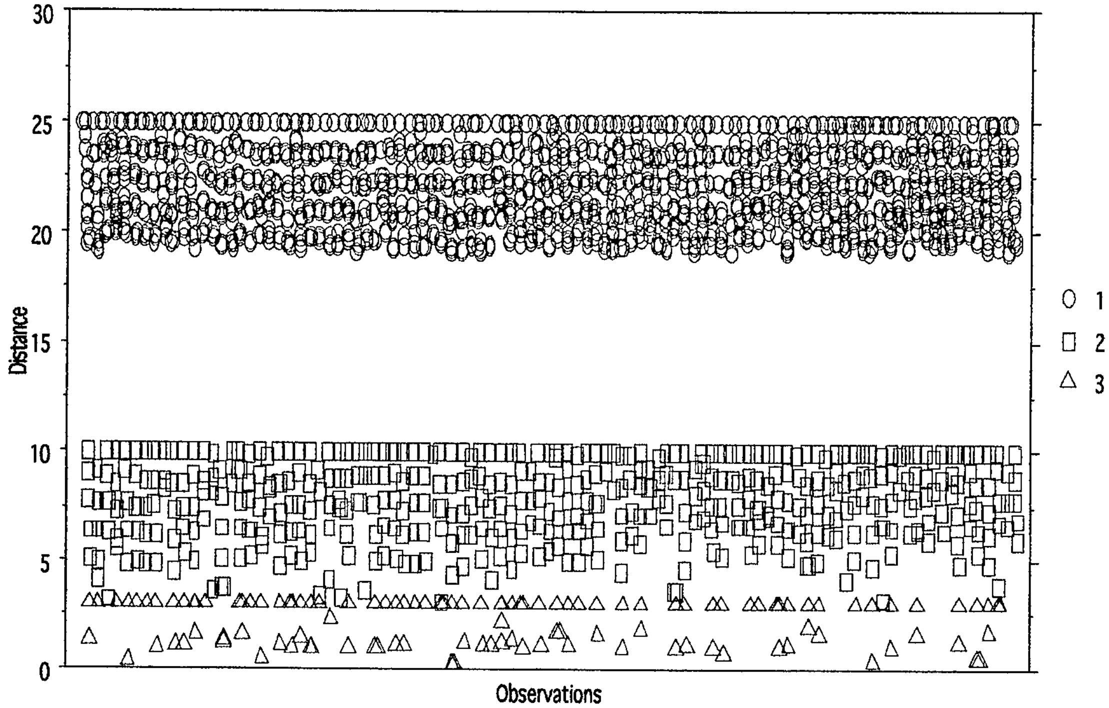

Figure 5-1: BaseLine Excursion - Interceptor Achieved Engagement Distance

the baseline stochastic excursion. Deterministically, all engagements would occur at the same engagement distance with the only variation occurring as a function of the reload capability of the tier. Every engagement should result in some damage to the missile and every missile should achieve some damage against the ship. This would be the same regardless of the excursion. Clearly in the stochastic example, the actual engagement range is distributed across the allowable engagement range and can be attributed to the interarrival times of the missiles, the reload times of the interceptors, and the probability of missile engagement. The damage to each missile and to the ship are discrete events that are independent of previous missile events. Intuitively, this is a more realistic simulation.

Table 5-1 demonstrates the differences between a purely deterministic

model and a stochastic model with respect to the ship defense scenario. In all cases the stochastic model can demonstrate each of the aspects of distribution in a manner the deterministic model cannot. In some cases such as the mean number of impacts against a ship, the values of the deterministic calculation will be significantly different from results obtained stochastically and in others the model will be insufficient to demonstrate a feature that has analytical significance.

Ability of Deterministic Model	Mean	Distribution of Engagement
Baseline	✔	✘
Reduced Interceptor Engagement Range	✔	✘
Reduced Interceptor Inventory	✘	✘
Reduced Interceptor P_{SSK}	✘	✘

Table 5-1: Ability of Deterministic Model vs. Stochastic

This is not meant to suggest that the Analyst's WorkBench as it is employed today is improperly used. Indeed, as a game replay mechanism, it is an easily manipulated tool. The ability to visualize the design of modeling methodologies makes a deterministic case very appealing. The "what if?" capability currently provided is essential for examining a question raised as the game is replayed. However, the ability to adequately examine a range of possibilities under a random environment is severely limited. A stochastic analysis better answers more specific questions such as, "How does changing my defense tier's engagement range affect my defensive capability?"

B. RECOMMENDATIONS

The stochastic enhancement developed in this thesis is easily incorporated into future version of the Analyst's WorkBench. MIGS hooks allowing the switch to multiple runs of the same experiment should be provided in the basic framework. AWSUM script language should be modified to allow the analyst to specify the statistical distribution from which a parameter can be drawn.

The developers of Analyst WorkBench may desire a visualization of the output from series of simulation runs. This would require the development of more specific data collection tools that would run at the framework level of the Analyst's WorkBench. Data would then be collected from each run of the excursion. The statistical means of the excursion would then be used as a parameter for the visualization run. The final step of an excursion would be the replay of the visualization run. The ability to look at the parameter value as the visualization is run needs to be modified to show variance and range of the parameter as a minimum. Additional software design is required to explore this possibility.

The scenario developed for this thesis is used for illustrative purposes only but may be useful in exploring the ability of a ship to defend itself. Further validation of the ship defense-in-depth process would result in a more realistic model that would be useful in studying defense-in-depth as a stochastic process. All ship defense capacity is a trade-off in ship offensive capability.

A recent thesis at the Naval Postgraduate School [Ref. 8] suggested that the employment of chaff and electronic countermeasures were essential in increasing the likelihood of successful ship defense. This observation might provide a rationale for reducing ship defensive weapon inventory and allowing more

offensive weapon inventory. The model developed for this thesis combined with the scenario of Reference 5 would provide useful trade-off analysis to determine a ship weapon inventory and operational employment to ensure maximum survivability while allowing mission capability.

APPENDIX. STOCHASTICS UNIT CODE

This chapter contains the Pascal code used to create the stochastic enhancement to the Analyst's WorkBench.

```
unit Stochastics;

{The beginning work for a stochastics unit to be incorporated in the Analyst Workbench
Framework}

{Created: 8 December 1993 by          LCDR Andrew W. Melton, USN}
{                                     Naval Postgraduate School}
{                                      Monterey, CA}
{Revision History: 3 Aug 1994}
{Incorporated Inverse Transforms from Law & Kelton "Simulation Modeling & Analysis" Chp8,
1991}
{29 July 1995 Added location, shape and scale parameters to several random variables}
{17 July changed to Unit to be added to AWB}

interface

uses
       Sane;
var
       index, n1: Integer;
       Zset, Tset: Longint;
       Zrng: array[1..100] of longint;

function fuzzyReal (r: REAL): REAL;{An easy test for model development}
procedure Randdf;
function   UniformN (x: integer; a, b: real): real;
function   ExponentialX (x: integer; lambda: real): real;
function   gammaX (x: integer): extended;
function   GammaN (x: integer; alpha, Beta: real): real;
function   Weibull (x: integer; alpha, Beta: real): real;
function   PoissonP (x: integer; lambda: real): real;
function   PoissonNextEvent (lambda: real; x: integer): real;
function   mErlang (x, m: integer; Beta: real): real;
function     NormalX (x: integer; mu, sigma: real): real; {Not a true Normal but close
                                                          enough}
function   CauchyX (x: integer; median, beta: real): real;
function   Rand (Stream: INTEGER): real; {Generate the next random number}
procedure  Randst (Zset: Longint; Stream: INTEGER);
function   Randgt (Stream: Integer): Longint;


implementation


type
       ZStatsPtr = ^ZStats;
       ZStats = record
           Zvalue: longint;
           Zindex: Integer;
```

```
        next: ZStatsPtr;
        end;

{This is a random number generator taken from Law & Kelton's book Simulation Modeling
and Analysis,1991}
{It is a PMMLCG supporting 100 streams of uniform random numbers}

procedure Randdf; {This may be something more appropriately stored in one of the globals
units}
      var
      i: INTEGER;
begin{set the seeds for all 100 streams}
```

```
Zrng[1] := 1973272912;    Zrng[2] := 281629770;     Zrng[3] := 20006270;
Zrng[4] := 1280689831;    Zrng[5] := 2096730329;    Zrng[6] := 1933576050;
Zrng[7] := 913566091;     Zrng[8] := 246780520;     Zrng[9] := 1363774876;
Zrng[10] := 604901985;    Zrng[11] := 1511192140;   Zrng[12] := 1259851944;
Zrng[13] := 824064364;    Zrng[14] := 150493284;    Zrng[15] := 242708531;
Zrng[16] := 75253171;     Zrng[17] := 1964472944;   Zrng[18] := 1202299975;
Zrng[19] := 233217322;    Zrng[20] := 1911216000;   Zrng[21] := 726370533;
Zrng[22] := 403498145;    Zrng[23] := 993232223;    Zrng[24] := 1103205531;
Zrng[25] := 762430696;    Zrng[26] := 1922803170;   Zrng[27] := 1385516923;
Zrng[28] := 76271663;     Zrng[29] := 413682397;    Zrng[30] := 726466604;
Zrng[31] := 336157058;    Zrng[32] := 1432650381;   Zrng[33] := 1120463904;
Zrng[34] := 595778810;    Zrng[35] := 877722890;    Zrng[36] := 1046574445;
Zrng[37] := 68911991;     Zrng[38] := 2088367019;   Zrng[39] := 748545416;
Zrng[40] := 622401386;    Zrng[41] := 2122378830;   Zrng[42] := 640690903;
Zrng[43] := 1774806513;   Zrng[44] := 2132545692;   Zrng[45] := 2079249579;
Zrng[46] := 78130110;     Zrng[47] := 852776735;    Zrng[48] := 1187867272;
Zrng[49] := 1351423507;   Zrng[50] := 1645973084;   Zrng[51] := 1997049139;
Zrng[52] := 922510944;    Zrng[53] := 2045512870;   Zrng[54] := 898585771;
Zrng[55] := 243649545;    Zrng[56] := 1004818771;   Zrng[57] := 773686062;
Zrng[58] := 403188473;    Zrng[59] := 372279877;    Zrng[60] := 1901633463;
Zrng[61] := 498067494;    Zrng[62] := 2087759558;   Zrng[63] := 493157915;
Zrng[64] := 597104727;    Zrng[65] := 1530940798;   Zrng[66] := 1814496276;
Zrng[67] := 536444882;    Zrng[68] := 1663153658;   Zrng[69] := 855503735;
Zrng[70] := 67784357;     Zrng[71] := 1432404475;   Zrng[72] := 619691088;
Zrng[73] := 119025595;    Zrng[74] := 880802310;    Zrng[75] := 176192644;
```

```
Zrng[76] := 1116780070;    Zrng[77] := 277854671;     Zrng[78] := 1366580350;
Zrng[79] := 1142483975;    Zrng[80] := 2026948561;    Zrng[81] := 1053920743;
Zrng[82] := 786262391;     Zrng[83] := 1792203830;    Zrng[84] := 1494667770;
Zrng[85] := 1923011392;    Zrng[86] := 1433700034;    Zrng[87] := 1244184613;
Zrng[88] := 1147297105;    Zrng[89] := 539712780;     Zrng[90] := 1545929719;
Zrng[91] := 190641742;     Zrng[92] := 1645390429;    Zrng[93] := 264907697;
Zrng[94] := 620389253;     Zrng[95] := 1502074852;    Zrng[96] := 927711160;
Zrng[97] := 364849192;     Zrng[98] := 2049576050;    Zrng[99] := 638580085;
Zrng[100] := 547070247;

end;{Randdf}

function Rand (Stream: INTEGER): real; {Generate the next random number}
       const
       B2E15 = 32768;
       B2E16 = 65536;
       MODULUS = 2147483647;
       MULTI1 = 24112;
       MULTI2 = 26143;

       var
       Hi15, Hi31, Low15, LowPrd, Ovflow, Zi: LONGINT;

begin {Rand}
       Zi := Zrng[Stream];
       Hi15 := Zi div B2E16;
       LowPrd := (Zi - Hi15 * B2E16) * MULTI1;
       Low15 := LowPrd div B2E16;
       Hi31 := Hi15 * MULTI1 + Low15;
       Ovflow := Hi31 div B2E15;
       Zi := (((LowPrd - Low15 * B2E16) - MODULUS) + Hi31 - Ovflow * B2E15) * B2E16 +
Ovflow;
       if Zi < 0 then
          begin
          Zi := Zi + MODULUS;
          end;
       Hi15 := Zi div B2E16;
       LowPrd := (Zi - Hi15 * B2E16) * MULTI2;
       Low15 := LowPrd div B2E16;
       Hi31 := Hi15 * MULTI2 + Low15;
       Ovflow := Hi31 div B2E15;
       Zi := (((LowPrd - Low15 * B2E16) - MODULUS) + Hi31 - Ovflow * B2E15) * B2E16 +
Ovflow;
       if Zi < 0 then
          begin
          Zi := Zi + MODULUS;
          end;
       Zrng[Stream] := Zi;
       Rand := (2 * (Zi div 256) + 1) / 16777216.0;
```

```
end;{Rand}

procedure AddToList (index: integer; Z: Longint; var Zlist: ZstatsPtr);
	var
	tempPtr: ZStatsPtr;
begin{AddToList}
	New(TempPtr);
	TempPtr^.ZIndex := index;
	TempPtr^.Zvalue := Z;
	TempPtr^.Next := Zlist;
	Zlist := TempPtr;
end;{AddToList}

procedure Randst (Zset: Longint; Stream: INTEGER);
	begin {Randst}

{Set the current Zrng for stream Stream to Zset}

	Zrng[Stream] := Zset;
	end; {Randst}


function Randgt (Stream: Integer): Longint;
begin {Randgt}

{Return the current Zrng for Stream}
Randgt := Zrng[Stream];
end; {Randgt}

function UniformN (x: integer; a, b: real): real;

	var
	tempU: real;

begin
	tempU := Rand(x);
	UniformN := a + (b - a) * tempU;
end;{UniformN}

function ExponentialX (x: integer; lambda: real): real;
{Specify the number stream from which to draw the seed}
{value and a real number for lambda. Returns a real from Exponential Dist }
	var
	tempU: real;

begin
	tempU := Rand(x);
	ExponentialX := -(ln(1 - tempU) / lambda);
end; {ExponentialX}



function gammaX (x: integer): extended;
{This only accepts an Integer input but in order to allow return of Numbers for Gammas}
{greater than Gamma of 20, I had to use extended numbers}
	var
```

```
        n: integer;
        TempProd: extended;

begin
        if x > 250 then
        begin
            writeln('This is too large a value for a Gamma Calculation ');
        end
        else
        begin
            tempProd := 1;
            for n := 1 to x - 1 do
            begin
                TempProd := TempProd * n;
            end;
                gammaX := TempProd;
        end;
end;{gammaX}

function GammaN (x: integer; alpha, Beta: real): real;
{This gives a number from a Gamma Distibution and is taken from pg 488-89 of L&K}
        var
        gamma, b, tempU1, tempU2, Y, V, Zu, a, q, theta, d, W: real;
        e: extended;
        success: boolean;

begin
        e := Exp(1);
        success := false;
        if Beta < 0 then
            begin
                writeln('Beta must be postive.'); {Error check}
            end
            else
                begin
                    if alpha < 0 then
                    begin
                    writeln(' Alpha value must be postive.'); {Error check}
                    end
                    else
                    begin
                        if alpha = 1 then
                        begin
                        gamma := ExponentialX(x, alpha);
                        end
                        else
                            begin
                            if (alpha > 0) and (alpha < 1) then
                            begin
                            b := (e + alpha) / e;
                                while not success do
                                begin
                                tempU1 := Rand(x) * b;
                                                {This P from pg 488 of Law & Kelton}
                                    if tempU1 > 1 then
                                    begin
```

```
                    Y := -ln((b - tempU1) / alpha);
                    tempU2 := Rand(x);
                        if ln(tempU2) <= ln(Y) * (alpha - 1) then
                        begin
                            gamma := Y;
                            success := true;
                        end; {first test}
                end
                else
                begin
                    Y := ln(tempU1) / alpha;
                    Y := exp(Y);
                    tempU2 := Rand(x);
                    if tempU2 <= exp(-Y) then
                        begin
                        gamma := Y;
                        success := true;
                        end; {second test}
                end;
            end; {while statement}
        end {0<alpha<1}
        else
        begin{alpha >1}
            a := 1 / sqrt(2 * alpha - 1);
            b := alpha - ln(4);
            q := alpha + 1 / a;
            theta := 4.5;
            d := 1 + ln(theta);
            while not success do
            begin
                tempU1 := Rand(x);
                tempU2 := Rand(x);
                V := a * ln(tempU1 / (1 - tempU1));
                Y := alpha * exp(V);
                Zu := sqr(tempU1) * tempU2;
                W := b + q * V - Y;
                if W + d - theta * Zu >= 0 then
                begin
                    gamma := Y;
                    success := true;
                end
                else
                begin
                    if W >= ln(Zu) then
                    begin
                        gamma := Y;
                        success := true;
                    end;
                end;
            end;{while}
        end; {alpha >1}
    end;
  end;
end;
GammaN := gamma * Beta;
```

```
end;{GammaN}

function Weibull (x: integer; alpha, Beta: real): real;

        var
        tempU: real;

begin
        tempU := Rand(x);
        Weibull := exp(ln(Beta) + ln(-ln(tempU)) / alpha);
end;{Weibull}

function PoissonP (x: integer; lambda: real): real;
{Returns a probability based on a Poisson distribution}
        var
        TempProd: extended;
        n: Integer;
        LnP: real;

begin
        TempProd := 1;
        if x = 0 then
{this prevents some divide by 0 errors - Still need to provide some guidance on inputing }
        begin  {negative integers}
            PoissonP := exp(-lambda);
        end
        else
        begin
            for n := 1 to x do
            begin
            TempProd := TempProd * n;
            end;
        LnP := x * ln(lambda) - lambda - ln(TempProd);
        PoissonP := exp(LnP);
        end;
end; {PoissonP}

function PoissonNextEvent (lambda: real; x: integer): real;
        var
        TempT: extended;

begin
        TempT := Ln(Rand(x)) / (-lambda);
        PoissonNextEvent := TempT;
end;



function mErlang (x, m: integer; Beta: real): real;

        var
        tempU: real;
        index: integer;

begin
        tempU := 1;
```

```
        for index := 1 to m do
            begin
            tempU := tempU * Rand(x);
            end;
        mErlang := -(Beta / m) * ln(tempU);
end;{mErlang}

function NormalX (x: integer; mu, sigma: real): real; {Not a true Normal but close
enough}
        var
        V1, V2, U1, U2, W, Y, X1, X2: real;

begin
        W := 1.5;
        while W > 1 do
            begin
            U1 := Rand(x);
            U2 := Rand(x);
            V1 := 2 * U1 - 1;
            V2 := 2 * U2 - 1;
            W := sqr(V1) + sqr(V2);
            end;
        Y := sqrt((-2 * ln(W)) / W);
        X1 := V1 * Y;
        X2 := V2 * Y;
        NormalX := X1 * sigma + mu; {This gives Normal~(mu,sigma^2)}
end;{NormalX}

function CauchyX (x: integer; median, beta: real): real;
        var
        tempU: real;

begin
        tempU := Rand(x);
        CauchyX := tan(pi * (tempU - 0.5)) + median / beta;
end; {CauchyX}


function  fuzzyReal (r: REAL): REAL;{Only use this for a quick test inAWB}

begin
        fuzzyReal := r;
end;    {fuzzyReal}



end.{Stochastics}
```

LIST OF REFERENCES

1. Ketcham, Ron, "Analyst WorkBench Overview," Unpublished, August 1993.

2. Law, A. M. and Kelton, W. D, *Simulation Modeling & Analysis*, pp 420-521, McGraw-Hill, Inc., 1991.

3. Symantec Corporation, *Think Pascal,* Symantic Corporation, Inc., San Jose, CA 1989.

4. Marsaglia, G. and Bray, T.A., "A Convenient Method for Generating Normal Variables,"pp. 260-264, *SIAM Rev*, 6 (1964).

5. Crawford, K. R., Hatton, M. T., and Melton, A. W., "Where Are the Littoral Warfare Fast-Attack Craft?," *Naval Institute Proceedings,* April 1995.

6. Ross, S. M., *Introduction to Probability Models*, pp. 389-394, Academic Press, Inc., 1989.

7. Abacus Concepts, *Statview,* Abacus Concepts, Inc., Berkeley, CA, 1992.

8. Schulte J. C., "An Analysis of the Historical Effectiveness of Anti-Ship Cruise Missiles in Littoral Warfare," Naval Postgraduate School Master's Thesis, Monterey, CA, September 1994.

BIBLIOGRAPHY

Bailey, Michael P., "Measuring Performance of Integrated Air Defense Networks Using Stochastic Networks," *Operations Research,* Vol. 40, Number 4., July-August 1992.

Mendenhall, William, Wackerly, Dennis D., and Scheaffer, Richard L., *Mathematical Statistics with Applications,* PWS-Kent Publishing Company, Boston, MA., 1990.

INTITIAL DISTRIBUTION LIST

No. of Copies

1. Defense Technical Information Center 2
   8725 John J. Kingman Rd., STE 0944
   Ft. Belvoir, VA 22060-6218

2. Dudley Knox Library 2
   Naval Postgraduate School
   411 Dyer Rd.
   Monterey, CA 93943-5101

3. Prof. Gary R. Porter, Code CC/Po. 1
   Naval Postgraduate School
   Monterey, CA 93943-5000

4. Prof. Michael P. Bailey, Code OR/Ba 1
   Naval Postgraduate School
   Monterey, CA 93943-5000

5. Ron Ketcham. 1
   Naval Air Warfare Center, Weapons Divsion
   Weapons Planning Group (C024304)
   China Lake CA 93555

6. Andrew W. Melton 1
   13915 Whetstone Manor Ct.
   Clifton, VA 20124-2530